E-ISSN: 2676-6159 | P-ISSN: 2538-5079

Journal of Decisions and Operations Research

www.journal-dmor.ir

J. Decis. Oper. Res. Vol. 6, No. 4 (2022) 536–552.

Paper Type: Original Article

Improved Genetic Algorithm with Diversity and Local Search for Uncapacitated Single Allocation Hub Location Problem

Mona Alizadeh Firozi[1], Vahid Kiani[2,*], Hossein Karimi[1]

[1] *Department of Industrial Engineering, Univeristy of Bojnord, Bojnord, Iran; h.karimi@ub.ac.ir.*

[2] *Department of Computer Engineering, Univeristy of Bojnord, Bojnord, Iran; v.kiani@ub.ac.ir.*

Citation:

Alizadeh Firozi, M., Kiani, V., & Karimi, H. (2022). Improved genetic algorithm with diversity and local search for uncapacitated single allocation hub location problem. *Journal of decisions and operations research*, *6*(4), 536-552.

Received: 21/05/2021 Reviewed: 12/07/2021 Revised: 01/08/2021 Accept: 05/09/2021

Abstract

Purpose: The purpose of this paper is to propose an improved genetic algorithm to solve the problem of Uncapacitated Single-allocation Hub Location. Previous methods have paid less attention to the diversity of population, and due to insufficient vairation in mutation operators, they perform well only in a few runs, and in other runs they are caught in the local optimum.

Methodology: The proposed method uses appropriate genetic operators to increase diversity of the population and performs local search around the best answer to exploit promising areas of the solution space. The use of hub mutation operators along with allocation mutation operators in the proposed algorithm has increased its exploration ability and effectiveness, which has led to discovery of the optimal answer in most runs for large size problems. Also, searching for the local neighborhood of the best answer made convergence faster and reduced the total running time for large instances.

Findings: Evaluation of the proposed method and base algorithm on the Australian Post (AP) dataset showed that the improvements increased efficiency of the genetic algorithm in achieving optimal solutions for problems as large as 200 nodes from 2% to more than 85%.

Originality/Value: This study showed that meta-heuristic algorithms and their improved versions are suitable methods for solving hub location problems in a short and limited time.

Keywords: Genetic algorithm, Meta-heuristic algorithms, Local search, Hub location.

Corresponding Author: h.karimi@ub.ac.ir

10.22105/DMOR.2021.272989.1325

Journal of Decisions and Operations Research

تصمیم‌گیری و تحقیق در عملیات

دوره ۶، شماره ۴، (۱۴۰۰)، ۵۵۲-۵۳۶

www.journal-dmor.ir

نوع مقاله: پژوهشی

یک الگوریتم ژنتیک بهبودیافته با گوناگونی و جستجوی محلی برای حل مسئله مکان یابی بدون ظرفیت هاب با تخصیص تکی

منا علیزاده فیروزی[۱]، وحید کیانی[۲*]، حسین کریمی[۱]

[۱]گروه مهندسی صنایع، دانشگاه بجنورد، بجنورد، ایران.

[۲]گروه مهندسی کامپیوتر، دانشگاه بجنورد، بجنورد، ایران.

چکیده

هدف: هدف این مقاله ارائه یک الگوریتم ژنتیک بهبودیافته برای حل مسئله مکان‌یابی بدون ظرفیت هاب با تخصیص تکی است. روش‌های پیشین حل مسئله کمتر به گوناگونی جواب‌ها در جمعیت توجه داشته‌اند و به دلیل عدم تنوع کافی در عملگرهای جهش تنها در برخی اجراها عملکرد مطلوبی دارند و در سایر اجراها در بهینه محلی گرفتار می‌شوند.

روش‌شناسی پژوهش: روش پیشنهادی از عملگرهای ژنتیک مناسب برای افزایش گوناگونی جمعیت و از جستجوی همسایگی محلی در اطراف بهترین جواب برای افزایش سرعت همگرایی استفاده می‌کند. استفاده از عملگرهای جهش هاب در کنار عملگرهای جهش تخصیص در الگوریتم پیشنهادی باعث کاوش بهتر فضای جستجو، افزایش کارایی و دستیابی به جواب بهینه در اکثر اجراها در مسائل با اندازه بزرگ شد. همچنین، جستجوی همسایگی محلی در اطراف بهترین جواب، باعث همگرایی سریع‌تر روش پیشنهادی شد و زمان حل مسئله را درمجموع برای مسائل بزرگ کاهش داد.

یافته‌ها: ارزیابی روش پیشنهادی و الگوریتم پایه روی مجموعه داده پست استرالیا (AP) نشان داد که بهبودهای انجام‌شده ضمن حفظ سرعت اجرا، کارایی الگوریتم ژنتیک را در دستیابی به جواب بهینه برای مسائلی به بزرگی ۲۰۰ گره از ۲% به بیش از ۸۵% افزایش می‌دهد.

اصالت/ارزش افزوده علمی: این مطالعه نشان داد که الگوریتم‌های فرا ابتکاری و نسخه‌های بهبودیافته آن‌ها می‌توانند روش‌های مناسبی برای حل انواع مسائل مکان‌یابی هاب در زمان کوتاه و محدود باشند.

کلیدواژه‌ها: الگوریتم ژنتیک، الگوریتم‌های فرا ابتکاری، جستجوی محلی، مکان‌یابی هاب.

۱- مقدمه

مسئله مکان‌یابی بدون ظرفیت هاب با تخصیص تکی[۱] یک مسئله مهم در شبکه‌های حمل‌ونقل است که در مسافرت هوایی، سیستم‌های پستی، سیستم‌های تحویل سریع، شبکه‌های کامپیوتری و مخابراتی و صنایع مشابه دیگر کاربرد دارد (بشیری و همکاران[۲]، ۲۰۱۱). در یک

---

[1] Uncapacitated Single Allocation Hub Location Problem (USAHLP)

[2] Bashiri et al.

* نویسنده مسئول

h.karimi@ub.ac.ir

10.22105/DMOR.2021.272989.1325

شبکه هاب، ترافیک بین گره‌های مختلف تجمیع شده و از طریق گره‌های خاصی به نام هاب و مسیرهای سریع بین آن‌ها انتقال می‌یابد. گره‌هایی از شبکه که هاب نیستند، گره غیرهاب نامیده می‌شوند. جمع‌آوری و توزیع بهینه ترافیک توسط گره‌های هاب و انتقال سریع ترافیک به کمک اتصال‌های بین هابی، باعث می‌شود یک شبکه حمل‌ونقل هاب نسبت به یک شبکه حمل‌ونقل غیرهابی صرفه‌جویی اقتصادی قابل‌توجهی را فراهم آورد. حکمت فر و پیشوایی[۱] (۲۰۰۹) انواع مسئله مکان‌یابی هاب، روش‌های حل و کاربردهای مختلف آن را به شکل گسترده‌ای بررسی کرده‌اند. در مسئله مکان‌یابی بدون ظرفیت هاب با تخصیص تکی که *Un-capacitated Single Allocation Hub Location Problem (USAHLP)* نامیده می‌شود، هر گره غیرهاب دقیقاً به یک گره هاب متصل می‌شود و هر هاب فاقد محدودیت ظرفیت است. هدف مسئله *USAHLP* کمینه‌سازی هزینه انتقال و هزینه تأسیس هاب‌ها است.

تا قبل از سال ۲۰۰۰ تحقیقات درزمینهٔ مسائل مکان‌یابی هاب بر تدوین و ارائه مدل‌های مختلف مسئله تمرکز داشتند، اما بعدازآن روش‌های حل توجه اکثر محققین را به خود جلب کرده‌اند (آلومور و کارا[۲]، ۲۰۰۸). در **جدول ۱** مهم‌ترین روش‌های حل مسئله مکان یابی بدون ظرفیت هاب با تخصیص تکی آمده است. در ادامه مرتبط‌ترین تحقیقات به مطالعه حاضر مرور خواهند شد. مسئله *USAHLP* ازجمله مسائل *NP-hard* است که حل دقیق آن در زمان چندجمله‌ای ممکن نیست. به همین دلیل با بزرگ شدن مسئله، حل دقیق به زمان زیاد و حافظه زیادی نیاز دارد. این موضوع را اولین بار اوکلی[۳] (۱۹۸۷) نشان داد و اولین کسی بود که دو روش حل ابتکاری را برای حل مسئله *USAHLP* پیشنهاد داد. اوکلی یک مدل ریاضی برنامه‌ریزی عدد صحیح درجه‌دو را برای مسئله *USAHLP* ارائه داد و سپس برای حل سریع مسئله از بررسی تعداد محدودی از انتساب‌های ممکن با در نظر گرفتن دو روش ابتکاری ساده استفاده کرد. در اولین روش ابتکاری، قاعده تخصیص به نزدیک‌ترین هاب[۴]، تنها انتساب‌هایی توسط اوکلی بررسی شدند که شامل انتساب هر گره غیرهاب به نزدیک‌ترین هاب بود. این روش ابتکاری در ادبیات موضوع قاعده مبتنی بر فاصله[۵] هم نامیده می‌شود. درروش ابتکاری دوم علاوه بر نزدیک‌ترین هاب، دومین نزدیک‌ترین هاب نیز بررسی می‌شد. ضمناً، اوکلی اولین مجموعه مسائل استاندارد برای حل مسئله *USAHLP* را ارائه داد که شامل اطلاعات ترافیک هوایی بین ۲۵ شهر بود و به‌اختصار مجموعه مسائل ترافیک هوایی (*CAB*[۶]) نامیده شد. پس‌ازآن ارنست و کریشنامورتی[۷] (۱۹۹۶) یک مدل خطی با تعداد کمتری متغیر ارائه دادند و ضمناً از جستجوی سردسازی شبیه‌سازی‌شده[۸] (*SA*) برای تخمین حد بالا درروش شاخه و کران استفاده کردند. علاوه به راین، ارنست و کریشنامورتی مجموعه مسائل پست استرالیا[۹] (*AP*) را نیز عرضه نمودند که مسائلی به بزرگی ۲۰۰ گره را شامل می‌شد و مرجعی برای ارزیابی روش‌های بعدی شد.

الگوریتم ژنتیک اولین بار توسط آبدینور هلم و ونکاتارامانان (۱۹۹۸) برای حل مسئله *USAHLP* مورداستفاده قرار گرفت و عملکرد آن با حل دقیق به کمک روش شاخه و کران مقایسه شد. پس‌ازآن، آبدینور هلم (۱۹۹۸) الگوریتم ژنتیک را با جستجوی ممنوعه (*TS*[۱۰]) درروشی تحت عنوان *GATS*[۱۱] ترکیب کرد و برای تقریباً تمام نمونه مسائل مجموعه داده ترافیک هوایی (*CAB*) به‌جز یکی از آن‌ها درزمانی قابل‌قبول به جواب بهینه دست‌یافت. در این رویکرد، از الگوریتم ژنتیک برای انتخاب هاب‌ها از بین کل گره‌ها و از جستجوی ممنوعه برای انتساب گره‌های غیرهاب به هاب‌ها بهره گرفته شد. نمایش جواب‌ها درروش آبدینورهلم به‌صورت بردارهای باینری بود که تحت عملگرهای ژنتیک تکامل می‌یافتند. تاپکواوغلو و همکاران (۲۰۰۵) یک‌راه حل جدید مبتنی بر الگوریتم ژنتیک برای حل این مسئله ارائه دادند و در مقایسه با الگوریتم *GATS* به جواب‌های بهتری در زمان کم‌تر بر روی مسائل بزرگ پست استرالیا (*AP*) دست یافتند. درروش پیشنهادی ایشان هر دو مسئله انتخاب گره‌های هاب و تخصیص گره‌های غیرهاب هم‌زمان توسط الگوریتم ژنتیک حل شد و نمایش جواب شامل دو بردار هاب و تخصیص برای هر جواب بود. تاپکواوغلو عملگرهای انتقال و جابجایی گره‌های غیرهاب را

[1] Hekmatfar and Pishvaee
[2] Alumur and Kara
[3] O'kelly
[4] Nearest-hub allocation rule
[5] Distance-based rule
[6] Civil Aeronautics Board (CAB)
[7] Ernst and Krishnamoorthy
[8] Simulated Annealing (SA)
[9] Australia Post (AP)
[10] Tabu Search (TS)
[11] Genetic Algorithm with Tabu Search (GATS)

به‌عنوان عملگر جهش مطرح کرد و برای انتخاب گره‌های هاب در جواب‌های اولیه نیز از یک رویکرد حریصانه تصادفی استفاده کرد که گره‌هایی با ترافیک بالا را به‌عنوان هاب ترجیح می‌دهد.

جدول ۱- مقایسه مطالعات انجام‌شده درزمینهٔ حل مسئله مکان‌یابی هاب.

Table 1- Comparison of reviewed studies on hub location problem.

منبع	روش حل	اندازه مسئله	نوآوری
اوکلی (۱۹۸۷)	برنامه‌ریزی عدد صحیح و هیوریستیک‌های ساده	۲۵	ارائه مدل برنامه‌ریزی عدد صحیح درجه‌دو، پیشنهاد دو روش هیوریستیک ساده.
ارنست و کریشنامورتی (۱۹۹۶)	سردسازی شبیه‌سازی‌شده و شاخه کران	۲۰۰	ارائه مدل برنامه‌ریزی عدد صحیح خطی، تخمین حد بالا با سردسازی شبیه‌سازی‌شده، روش شاخه و کران برای حل مسئله.
آبدینورهلم و ونکاتارامانان[1] (۱۹۹۸)	الگوریتم ژنتیک	۸۰	عملگر تقاطع تک نقطه‌ای، عملگر جهش معکوس سازی بیت.
آبدینورهلم[2] (۱۹۹۸)	ترکیب الگوریتم ژنتیک و جستجوی ممنوعه	۲۰۰	استفاده از جستجوی ممنوعه برای تخصیص.
تاپکواوغلو و همکاران[3] (۲۰۰۵)	الگوریتم ژنتیک	۲۰۰	عملگرهای انتقال و جابجایی گره‌های غیرهاب.
فیلیپوویچ و همکاران[4] (۲۰۰۹)	الگوریتم ژنتیک	۲۰۰	بهبود با انتخاب تورنومنت، حافظه نهان، اجتناب از جواب‌های تکراری.
سیلوا و کانها[5] (۲۰۰۹)	جستجوی ممنوعه چندشروعه و دوسطحی	۴۰۰	توجه به ترافیک و فرکانس گره‌های هاب، استفاده از جستجوی ممنوعه در دو سطح هاب و غیرهاب.
ماریک و همکاران[6] (۲۰۱۳)	ترکیب الگوریتم ژنتیک و جستجوی محلی	۹۰۰	ترکیب الگوریتم ژنتیک با جستجوی محلی.
بیلی و همکاران[7] (۲۰۱۳)	الگوریتم توده ذرات	۲۰۰	الگوریتم توده ذرات گسسته، مکانیزم نویز و لرزش برای رهایی از کمینه محلی.
ابیضی ثانی و قنبری[8] (۲۰۱۶)	جستجوی ممنوعه	۴۰۰	عملگرهای افزودن هاب، حذف هاب، جایگزینی هاب، عملگر جابجایی غیرهاب.
ربانی و همکاران[9] (۲۰۱۷)	الگوریتم ژنتیک	۲۰۰	در نظر گرفتن هزینه‌های جدید در تابع هزینه ازجمله هزینه ایمنی، سلامت و هزینه محیطی.
اکبری پور و همکاران[10] (۲۰۱۷)	الگوریتم جستجوی پراکنده	۲۰۰	تحلیل فضای جستجو با کمک آنتروپی، استفاده از الگوریتم جستجوی پراکنده.
کاروالیو و همکاران[11] (۲۰۱۷)	جستجوی چندشروعه	۲۰۰	شروع مجدد با مکانیزم آشفته سازی، جستجوی محلی.

---

1 Abdinnour-Helm and Venkataramanan

2 Abdinnour-Helm

3 Topcuoglu et al.

4 Filipović et al.

5 Silva and Cunha

6 Marić et al.

7 Bailey et al.

8 Abyazi-Sani and Ghanbari

9 Rabbani et al.

10 Akbaripour et al.

11 Carvalho et al.

گوآن و همکاران[1] (b۲۰۱۸)	جستجوی محلی چند شروعه	۹۰۰	مکانیزم آشفته سازی برای شروع مجدد، تولید مجموعه جواب‌های همسایه امیدوارکننده.
گوآن و همکاران (a۲۰۱۸)	جستجوی ممنوعه احتمالاتی مبتنی بر یادگیری	۹۰۰	مکانیزم یادگیری برای اجتناب از هاب‌های ضعیف، جستجوی ممنوعه احتمالاتی برای تولید همسایه‌ها.
غفاری نسب و کارا[2] (۲۰۱۹)	الگوریتم تجزیه بندرز	۲۰۰	طرح تفکیک برش، تولید برش‌های بهینه قوی، استفاده از تنها یک درخت جستجو.

فیلیپوویچ و همکاران (۲۰۰۹) یک الگوریتم ژنتیک مجهز به عملگر انتخاب تورنومنت، عملگر تقاطع تک نقطه‌ای و عملگر جهش خاص ارائه کردند که باعث تغییر یک ژن در بخش هاب یا بخش تخصیص جواب می‌شد. فیلیپوویچ از نخبه‌گرایی با ۱۰۰ جواب نخبه، عدم جهش روی بیت‌های تثبیت‌شده در هنگام جهش، حذف جواب‌های تکراری از جمعیت، محدود کردن تعداد جواب‌هایی که برازندگی یکسان دارند و بهبود سرعت با کمک حافظه نهانی به‌اندازه ۵۰۰۰ جواب نیز استفاده کرده است. این روش که با پیچیدگی‌های فنی بسیاری همراه است توانست در زمان کمتری نسبت به روش تاپکواوغلو نمونه مسائل مجموعه داده پست استرالیا (*AP*) را حل کرده و در حداقل یکی از بیست اجرا برای تمامی مسائل به جواب بهینه دست یابد؛ اما نتایج فیلیپوویچ نشان می‌دهد که برای مسائلی با ۴۰ گره یا بیشتر در بعضی اجراها، روش پیشنهادی تنها به جواب زیر بهینه دست پیداکرده است.

جستجوی ممنوعه چند شروعه[3] و جستجوی ممنوعه دوسطحی برای حل مسئله *USAHLP* استفاده شده است (سیلوا و کانها، ۲۰۰۹). سیلوا در جستجوی ممنوعه چند شروعه برای انتخاب گره‌های هاب در جواب اولیه برای هر گره یک احتمال انتخاب در نظر گرفته و انتخاب گره‌های هاب را به‌تصادف انجام داده است، سپس از روش جستجوی ممنوعه برای بهینه‌سازی تخصیص‌ها استفاده کرده است. سه رویکرد برای محاسبه احتمال انتخاب توسط سیلوا مطرح‌شده است: احتمال انتخاب یکسان برای همه گره‌ها، احتمال انتخاب بیشتر برای گره‌هایی با ترافیک بیشتر، ترکیب ترافیک و تعداد دفعاتی که قبلاً آن گره به‌عنوان هاب انتخاب‌شده است برای اجتناب از بررسی هاب‌های تکراری. سپس جستجوی ممنوعه بر روی تخصیص گره‌های غیرهاب انجام‌شده و از عملگر انتقال گره غیرهاب از هاب فعلی به یک هاب دیگر به‌عنوان عملگر جستجو استفاده‌شده است. بدون تغییر گره‌های هاب و تنها با تغییر تخصیص گره‌های غیرهاب، احتمال بهبود جواب کم است. به همین دلیل، سیلوا در یک رویکرد دیگر از جستجوی ممنوعه دوسطحی استفاده کرده است که سطح اول به انتخاب گره‌های هاب و سطح دوم به تخصیص گره‌های غیرهاب می‌پردازد. آزمایش‌های انجام‌شده توسط سیلوا بر روی نمونه مسائل پست استرالیا (*AP*) نشان داد که جستجوی ممنوعه دوسطحی نسبت به جستجوی ممنوعه چند شروعه تک سطحی عملکرد بهتری دارد. سیلوا همچنین مجموعه مسائل پست استرالیا (*AP*) را توسعه داد و مسائلی به بزرگی ۳۰۰ و ۴۰۰ گره را نیز به آن افزود.

ماریک و همکاران (۲۰۱۳) از ترکیب الگوریتم ژنتیک با جستجوی محلی برای حل مسئله مکان‌یابی بدون ظرفیت هاب با تخصیص تکی استفاده کرده‌اند و از ترکیب هم‌زمان دو روش جستجوی محلی متفاوت بهره برده‌اند. اگرچه جستجوی محلی باعث افزایش سرعت روش پیشنهادی ماریک و دستیابی به جواب بهینه برای مسائل بزرگ‌تر شده است، اما نتایج ماریک نشان می‌دهد که این روش نیز در بعضی اجراها برای مسائلی با اندازه بزرگ‌تر از ۴۰ گره در مجموعه مسائل پست استرالیا از دستیابی به جواب بهینه ناتوان است. ماریک همچنین به توسعه مجموعه داده پست استرالیا پرداخته و نمونه مسائلی با اندازه ۵۲۰، ۶۰۰، ۷۲۰، ۸۰۰ و ۹۰۰ گره را نیز به آن افزوده است. ابیضی ثانی و قنبری (۲۰۱۶) یک روش جستجوی ممنوعه بهبودیافته را برای حل مسئله *USAHLP* مطرح کرده‌اند که از عملگرهای افزودن هاب، حذف هاب و تعویض هاب با غیرهاب برای دست‌کاری گره‌های هاب و از عملگر جابجایی غیرهاب برای تغییر تخصیص‌ها استفاده می‌کند. در عملگر جابجایی غیرهاب یک گره غیرهاب از هاب فعلی‌اش به هاب دیگری تخصیص می‌یابد. ابیضی ثانی با ساخت ماتریس نزدیک‌ترین همسایه‌ها (*R*)، عملگرهای کامل‌تر، لیست کاندیداها و لیست ممنوعه توانسته است در زمان

[1] Guan et al.
[2] Ghaffarinasab and Kara
[3] Multi-start Tabu Search (MSTS)

کمتری نسبت به روش‌های قبلی به جواب بهینه مسائل پست استرالیا (AP) به بزرگی ٤٠٠ گره دست یابد. ربانی و همکاران (٢٠١٧) هزینه‌های جدیدی شامل هزینه سلامت، هزینه ایمنی، هزینه انرژی و هزینه پرسنل را در مسئله *USAHLP* مطرح کرده و از الگوریتم ژنتیک برای حل سریع مسئله استفاده کرده‌اند. گوآن و همکاران (٢٠١٨) از جستجوی محلی تکراری چند شروعی[1] برای حل مسئله *USAHLP* استفاده کرده‌اند. درروش پیشنهادی گوآن ابتدا یک جواب اولیه با رویکردی حریصانه ساخته می‌شود، سپس جواب فعلی در یک فرآیند تکراری با کمک جستجوی محلی تکراری بهبود می‌یابد. گوآن همچنین از یک مکانیزم آشفته سازی[2] برای فرار از بهینه محلی استفاده کرده است. گوآن فرآیند جستجوی محلی را تنها بر روی جواب‌های امیدوارکننده‌ای انجام داده است که در همسایگی جواب فعلی قرار دارند. برای به دست آوردن جواب‌های همسایه از عملگر تعویض یک هاب با غیرهاب استفاده‌شده است. پس از شناسایی همسایه‌های امیدوارکننده، تخصیص گره‌های غیرهاب در یک فرآیند جستجوی محلی اصلاح‌شده است. گوآن در صورت عدم بهبود جواب فعلی پس از تعداد مشخصی تلاش، به‌جای شروع مجدد از یک جواب تصادفی جدید، از عملگر آشفته سازی بر روی جواب فعلی برای شروع مجدد استفاده کرده است. روش پیشنهادی گوآن توانسته است مسائلی تا اندازه ٩٠٠ گره را در زمان کمتری نسبت به روش‌های قبلی حل کند و برای تمام نمونه مسائل بررسی‌شده به مقدار صفر برای نرخ فاصله تا جواب بهینه[3] در تمام اجراها دست یابد. در کنار این‌ها، الگوریتم تجزیه بندرز (غفاری نسب و کارا، ٢٠١٩)، جستجوی ممنوعه احتمالاتی[4] (گوآن و همکاران، a٢٠١٨)، جستجوی محلی چند شروعه (کاروالیو و همکاران، ٢٠١٧) و الگوریتم توده ذرات[5] (بیلی و همکاران، ٢٠١٣) نیز برای حل مسئله *USAHLP* استفاده‌شده است.

مسئله مکان‌یابی بدون ظرفیت هاب با تخصیص تکی قبلاً در (تاپکواوغلو و همکاران، ٢٠٠٥)، (فیلیپوویچ و همکاران، ٢٠٠٩)، (ماریک و همکاران، ٢٠١٣) با کمک الگوریتم ژنتیک حل‌شده است. بررسی نتایج آن‌ها نشان می‌دهد که هر سه روش برای مسائلی با اندازه بزرگ‌تر از ٤٠ گره در بعضی اجراها به جواب زیر بهینه همگرا شده‌اند و از دستیابی به جواب بهینه در تمام اجراها ناتوان بوده‌اند. علاوه بر این، پیچیدگی‌های فنی روش پیشنهادی فیلیپوویچ تنها باعث افزایش سرعت حل مسئله شده‌اند و استفاده از این روش را برای سایر محققین دشوار کرده‌اند. بر همین اساس، در این مطالعه یک الگوریتم ژنتیک بهبودیافته مطرح خواهد شد که توسعه‌ای بر روش پیشنهادی تاپکواوغلو در (تاپکواوغلو و همکاران، ٢٠٠٥) است. روش پیشنهادی ما که حاصل افزودن عملگرهای جهش هاب و عملگر جستجوی محلی به روش تاپکواوغلو است، قادر است برای مسائل کوچک در تمام اجراها و برای مسائل بزرگ در اکثر اجراها به جواب بهینه نمونه مسائل بررسی‌شده دست یابد. مقایسه نتایج الگوریتم پیشنهادی بهبودیافته با الگوریتم تاپکواوغلو در (تاپکواوغلو و همکاران، ٢٠٠٥) حاکی از افزایش توانایی روش پیشنهادی در دست‌یابی به جواب بهینه برای نمونه مسائل پست استرالیا (AP) است.

٢- بیان ریاضی مسئله

مجموعه گره‌های $N=\{1,2,...,n\}$ را که شامل گره‌های هاب و غیرهاب است در یک شبکه حمل‌ونقل در نظر بگیرید. برای هر جفت گره (i,j) میزان ترافیکی که باید از گره i به گره j منتقل شود را با W_{ij} نشان می‌دهیم. فاصله هر دو گره را نیز با d_{ij} نشان خواهیم داد. هر یک از گره‌های ١ تا n می‌توانند به‌عنوان هاب انتخاب شوند. برای تأسیس یا راه‌اندازی هر مرکز هاب در مکان نامزد k هزینه تأسیس F_k را در نظر بگیرید. مسئله شامل دو بخش است: انتخاب بهینه هاب‌ها از بین کل گره‌ها (مکان‌یابی) و تخصیص گره‌های باقی‌مانده به‌عنوان غیرهاب به هاب‌ها (تخصیص). هر گره غیرهاب دقیقاً به یک هاب تخصیص می‌یابد به‌گونه‌ای که هزینه کلی انتقال ترافیک شبکه و هزینه تأسیس هاب‌ها به حداقل برسد. فرض کنید اگر گره غیرهاب i به گره هاب j تخصیص یابد، متغیر تصمیم $Zij \in \{0,1\}$ برابر با یک باشد و در غیر این صورت صفر باشد. همچنین، برای گره هاب k مقدار متغیر تصمیم $Z_{kk} = 1$ است و نشان می‌دهد که گره k به‌عنوان هاب انتخاب‌شده است. لذا انتخاب گره‌های هاب و تخصیص گره‌های غیرهاب به هاب‌ها از طریق متغیر تصمیم Z_{ij} محقق خواهد شد.

[1] Multi-start Iterated Local Search

[2] Perturbation

[3] Average Percentage Gap (APG)

[4] Probabilistic Tabu Search

[5] Particle Swarm Optimization (PSO)

هر جریان[1] از i تا j انتقال یک محموله از گره i به گره j را نشان می‌دهد که شامل سه مؤلفه هزینه جمع‌آوری[2]، هزینه انتقال[3] و هزینه توزیع[4] است. فرض کنید گره i به هاب k و گره j به هاب l متصل باشد. هزینه جمع‌آوری نشان‌دهنده هزینه انتقال محموله از گره مبدأ i به هاب k است. هزینه انتقال نشان‌دهنده هزینه انتقال محموله بین هاب‌ها است. هزینه توزیع نشان‌دهنده هزینه انتقال محموله از هاب l به گره مقصد j است. هرکدام از این فعالیت‌ها دارای هزینه مشخصی است که متناسب با مقدار محموله، مسافت دو گره و ضریب هزینه انتقال بین آن دو نوع گره تعیین می‌شود. به‌عبارت‌دیگر، با توجه به ماهیت متفاوت لینک‌های بین‌هابی و لینک‌های متصل به گره‌های غیرهاب، ضریب هزینه انتقال برای مراحل جمع‌آوری، انتقال و توزیع متفاوت در نظر گرفته می‌شود. فرض کنید ضریب هزینه جمع‌آوری را با α و ضریب هزینه انتقال بین هابی را با γ و ضریب هزینه توزیع را با δ نشان دهیم. معمولاً به دلیل تخفیف ناشی از انتقال حجمی کالا و مسافر در حالت‌های حمل‌ونقل باصرفه اقتصادی، مقدار ضریب هزینه γ بسیار کمتر از α یا δ است.

بر این اساس، مسئله مکان‌یابی بدون ظرفیت هاب با تخصیص تکی (*USAHLP*) را می‌توان به شکل زیر تعریف کرد (اوکلی، ۱۹۸۷) و (ارنست و کریشنامورتی، ۱۹۹۶):

$$\sum_{i\in N}\sum_{k\in N}\sum_{l\in N}\sum_{j\in N} W_{ij}\left(\alpha d_{ik} Z_{ik} + \gamma d_{kl} Z_{ik} Z_{jl} + \delta d_{jl} Z_{jl}\right). \tag{۱}$$

$s.t.$

$$\sum_{k\in N} Z_{ik} = 1, \qquad \forall i \in N. \tag{۲}$$

$$Z_{ik} \le Z_{kk}, \qquad \forall i,k \in N. \tag{۳}$$

$$Z_{ik} \in \{0,1\}, \qquad \forall i,k \in N. \tag{۴}$$

قید (۲) به همراه **قید (۴)** باعث می‌شوند هر گره غیرهاب فقط به یک گره هاب تخصیص یابد. **قید (۳)** نیز از تخصیص گره هاب یا غیرهاب به گره غیرهاب جلوگیری می‌کند.

۳- الگوریتم ژنتیک پیشنهادی

در این مقاله یک الگوریتم ژنتیک برای حل مسئله مکان‌یابی بدون ظرفیت هاب با تخصیص تکی (*USAHLP*) ارائه می‌شود که مجهز به عملگرهای جهش مناسب و جستجوی محلی است. **شکل ۱** مراحل روش پیشنهادی را به‌صورت یک فلوچارت نمایش می‌دهد. ابتدا جمعیت اولیه به‌صورت تصادفی تولید می‌شود، طوری که بخش اعظم جمعیت از تعداد کمی هاب و بخش کمتری از جمعیت از تعداد زیادی هاب برخوردار باشند. سپس برازندگی هر راه‌حل در جمعیت ارزیابی می‌شود. برازندگی هر راه‌حل در جمعیت با کمک‌هزینه آن راه‌حل در **رابطه (۱)** محاسبه می‌شود و برابر با یک تقسیم‌بر هزینه راه‌حل است. در مرحله بعدی بعضی از افراد نسل فعلی به کمک عملگر انتخاب چرخ رولت به‌عنوان والدین نسل بعدی انتخاب می‌شوند. سپس هر دو والد با یک احتمال با کمک عملگر تقاطع تک نقطه‌ای باهم ترکیب می‌شوند و دو جواب جدید به وجود می‌آورند. سپس، بعضی اعضای نسل جدید بر اساس احتمال جهش توسط عملگرهای جهش دست‌کاری می‌شوند. درنهایت باسیاست نخبه‌گرایی بهترین راه‌حل نسل فعلی به نسل بعدی کپی می‌شود. الگوریتم تا برقراری شرایط توقف ادامه می‌یابد. شرط توقف اتمام تکرارها یا عدم بهبود در حداقل ۵۰ تکرار متوالی در نظر گرفته‌شده است.

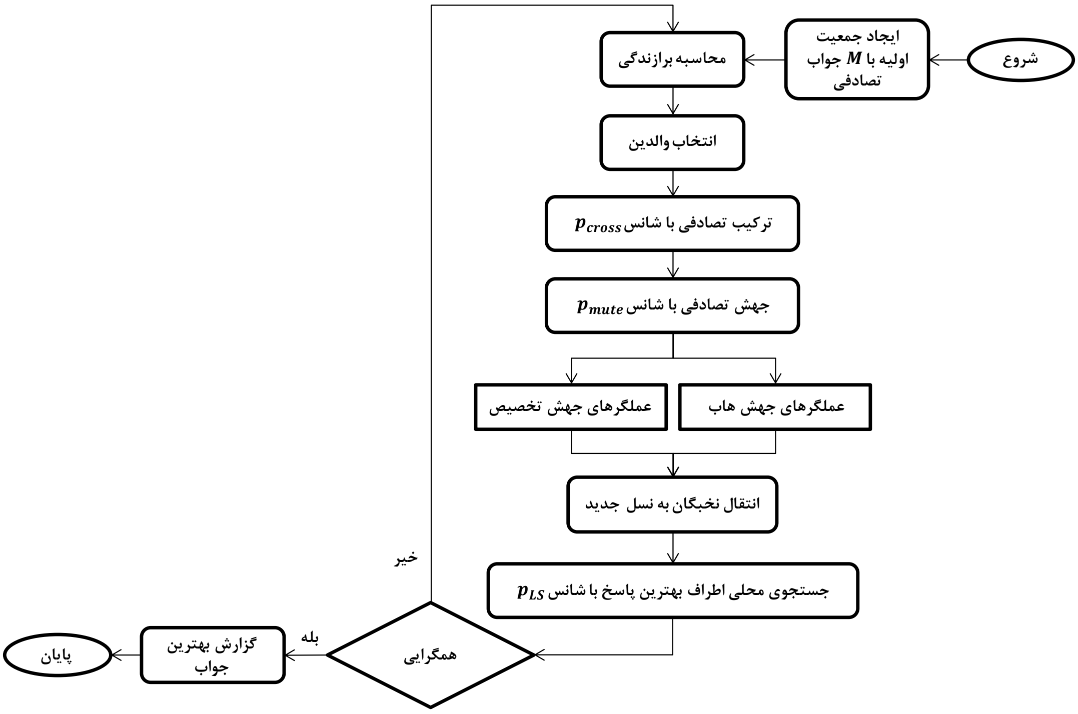
در این بخش، جزئیات راه‌حل پیشنهادی خود را با تأکید بر بخش‌های جدید نسبت به روش پیشنهادی (تاپکواوغلو و همکاران، ۲۰۰۵) ارائه می‌دهیم. تمام مراحل و جزئیات روش پیشنهادی این مقاله مشابه روش پیشنهادی (تاپکواوغلو و همکاران، ۲۰۰۵) است، اما عملگرهای جهش هاب و عملگر جستجوی محلی به آن افزوده‌شده‌اند. نوآوری‌های روش پیشنهادی عبارت‌اند از:

---

[1] Flow
[2] Collection cost
[3] Transfer cost
[4] Distribution cost

۱- عملگرهای جهش هاب به روش پیشنهادی افزوده‌شده‌اند تا بر انتخاب گره‌های هاب نیز تأثیر بگذراند. این عملگرها توانایی اکتشاف الگوریتم پیشنهادی را افزایش می‌دهند.

۲- درروش پیشنهادی این مقاله، الگوریتم ژنتیک با جستجوی محلی ترکیب‌شده است تا سرعت همگرایی و دستیابی به جواب بهینه بیشتر شود.

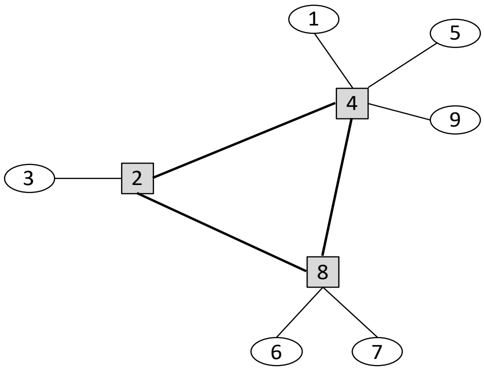


شکل ۱- فلوچارت الگوریتم ژنتیک پیشنهادی.

Figure 1- Flowchart of proposed genetic algorithm.

۳-۱- نمایش رشته

الگوریتم ژنتیک با جمعیتی از جواب‌ها کار می‌کند. درروش پیشنهادی مقاله حاضر مشابه (تاپکواوغلو و همکاران، ۲۰۰۵)، هر جواب در جمعیت از دو آرایه بردار هاب (*Hub Array*) و بردار تخصیص (*Assign Array*) تشکیل‌شده است. طول این بردارها برابر با تعداد گره‌های شبکه است. بردار هاب یک بردار باینری است که مقدار یک در آن نشانگر هاب بودن گره و مقدار صفر بیانگر غیرهاب بودن گره است. بردار تخصیص یک بردار از اعداد صحیح است که تخصیص گره‌های غیرهاب به هاب‌ها را نشان می‌دهد. اگر گره غیرهاب i به گره هاب k تخصیص یابد، خانه شماره i در بردار تخصیص مقدار k دارد. علاوه بر این، هر هاب در بردار تخصیص به خودش تخصیص داده می‌شود. **شکل ۲** نمونه‌ای از یک جواب مسئله و شبکه متناظر با آن را برای ۹ گره نمایش می‌دهد. در این مثال، گره‌های ۲، ٤ و ۸ هاب هستند. گره‌های غیرهاب ۱، ۵ و ۹ به هاب ٤، گره ۳ به هاب ۲ و گره ٦ و ۷ به هاب ۸ تخصیص داده‌شده‌اند.



نمایش جواب	1	2	3	4	5	6	7	8	9
بردار هاب	0	1	0	1	0	0	0	1	0
بردار تخصیص	4	2	2	4	4	8	8	8	4

شکل ۲- نمایش جواب در الگوریتم ژنتیک پیشنهادی و شبکه حمل‌ونقل متناظر با آن.

Figure 2- Representation of a solution in proposed genetic algorithm and corresponding transportation network.

۳-۲- عملگر تقاطع

در این مقاله، مشابه روش پیشنهادی (تاپکواوغلو و همکاران، ۲۰۰۵) از عملگر تقاطع تک نقطه‌ای روی هر دو آرایه بردار هاب و بردار تخصیص برای تولید فرزندان استفاده شد؛ اما این کار می‌تواند باعث تولید جواب‌های نشدنی شود که در آن‌ها بردار تخصیص با بردار هاب سازگاری ندارد. در مورد این جواب‌ها از یک فرآیند تعمیر استفاده شد که در آن اولاً هر هاب به خودش تخصیص داده شد و ثانیاً برای گره‌های غیرهابی که به گره‌ای تخصیص داده‌شده‌اند که دیگر هاب نیست، از نزدیک‌ترین هاب در جواب جدید استفاده شد. **شکل ۳** با یک مثال نحوه عملکرد عملگر تقاطع را نشان می‌دهد.

در **شکل ۳**، در ابتدا گره‌های ۳ و ۷ هاب‌های والد اول و گره‌های ٤ و ۸ هاب‌های والد دوم هستند. با اعمال عملگر تقاطع بر روی این دو والد، فرزند اول و فرزند دوم تولیدشده‌اند. در فرزند اول، گره‌های ۳ و ۸ هاب هستند و در فرزند دوم گره‌های ٤ و ۷ هاب هستند. استفاده از عملگر تقاطع باعث شده است تخصیص گره‌های ۱، ٦ و ۹ در فرزند اول معتبر نبوده و نیازمند تعمیر باشد، زیرا گره‌های ٤ و

والد اول	1	2	3	4	5	6	7	8	9
بردار هاب	0	0	1	0	0	0	1	0	0
بردار تخصیص	7	3	3	3	3	7	7	3	3

والد دوم	1	2	3	4	5	6	7	8	9
بردار هاب	0	0	0	1	0	0	0	1	0
بردار تخصیص	4	4	8	4	4	4	8	8	4

عملگر تقاطع:

فرزند اول	1	2	3	4	5	6	7	8	9
بردار هاب	0	0	1	0	0	0	0	1	0
بردار تخصیص	7	3	3	3	3	4	8	8	4

فرزند دوم	1	2	3	4	5	6	7	8	9
بردار هاب	0	0	0	1	0	0	1	0	0
بردار تخصیص	4	4	8	4	4	7	7	3	3

فرآیند تعمیر:

فرزند اول	1	2	3	4	5	6	7	8	9
بردار هاب	0	0	1	0	0	0	0	1	0
بردار تخصیص	8	3	3	3	3	8	8	8	3

فرزند دوم	1	2	3	4	5	6	7	8	9
بردار هاب	0	0	0	1	0	0	1	0	0
بردار تخصیص	4	4	7	4	4	7	7	4	7

۷ دیگر در فرزند اول هاب نیستند. برای تعمیر این تخصیص‌ها، هر یک از گره‌های غیرهاب ۱، ٦ و ۹ به نزدیک‌ترین هاب در جواب جدید تخصیص داده‌شده است. فرآیند تعمیر مشابهی برای بردار تخصیص فرزند دوم نیز انجام‌شده است.

شکل ۳- عملگر تقاطع تک نقطه‌ای و فرآیند تعمیر.
Figure 3- One-point cross-over operator and repair procedure.

۳-۳- عملگر جهش

در یک جمعیت طبیعی، جهش‌های ژنتیکی در فرزندان باعث تولید افرادی می‌شود که بعضی خصوصیات آن‌ها با هر دو والدشان متفاوت باشد. در الگوریتم ژنتیک نیز عملگر جهش به‌صورت تصادفی روی بعضی فرزندان اعمال‌شده و باعث تغییر تصادفی تعدادی از ژن‌ها می‌شود. در این مقاله، از چند عملگر جهش مختلف روی افراد نسل جدید استفاده شد تا احتمال فرار از کمینه‌های محلی و گوناگونی جمعیت افزایش یابد. عملگرهای جهش درروش پیشنهادی عبارت‌اند از:

(۱) انتقال گره غیرهاب (*shift spoke*): این عملگر جهش یک گره غیرهاب را انتخاب می‌کند و آن را به هاب دیگری که به‌طور تصادفی انتخاب‌شده است، تخصیص می‌دهد. اگر فقط یک هاب در رشته وجود داشته باشد، این عملگر جهش اعمال نمی‌شود. **شکل ٤ (الف)** اثر عملگر جهش انتقال گره غیرهاب را با یک مثال نشان می‌دهد. در این مثال، گره غیرهاب ۹ که قبلاً به هاب ٤ تخصیص داشته است، در تخصیص جدید به هاب ۲ انتقال داده‌شده است.

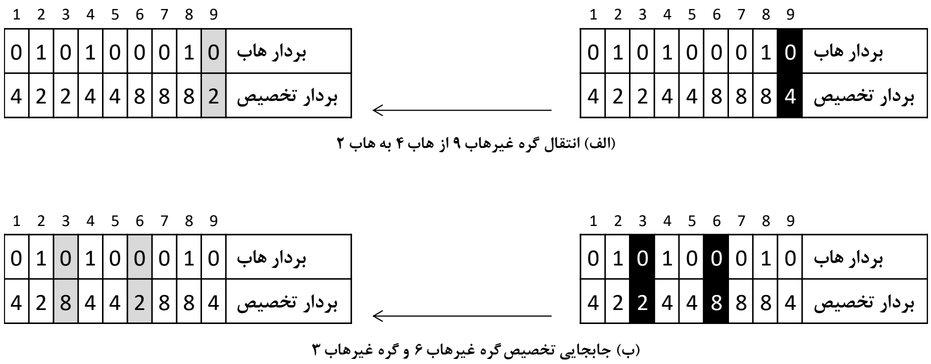
(۲) جابجایی گره‌های غیرهاب (*exchange spokes*): دو گره غیرهاب را به‌طور تصادفی انتخاب می‌کند و تخصیص آن‌ها را جابجا می کند. به‌عنوان نمونه در **شکل ٤ (ب)** تخصیص گره غیرهاب ٦ با تخصیص گره غیرهاب ۳ جابجا شده است. ازآنجاکه حداقل به یک جفت گره هاب و یک جفت گره غیرهاب نیاز داریم، اگر فقط یک گره هاب یا فقط یک گره غیرهاب وجود داشته باشد، عملگر جابجایی قابل‌اجرا نیست.

(۳) افزودن هاب (*add hub*): یک گره غیرهاب را به‌تصادف انتخاب و آن را هاب می‌کند. برای بقیه گره‌های غیرهاب، به ازای هر گره غیرهاب فاصله تا هاب جدید و هاب فعلی خودش چک می‌شود و آن گره غیرهاب به هاب نزدیک‌تر تخصیص می‌یابد. به‌عنوان‌مثال، در **شکل ۵ (الف)** گره‌های ۲، ٤ و ۸ هاب هستند. می‌خواهیم گره ٦ را نیز به هاب تبدیل کنیم؛ بنابراین در بردار هاب بیت متناظر با گره ٦ روی یک تنظیم می‌شود. در بردار تخصیص، گره ٦ باید به خودش تخصیص یابد. همچنین برای هر گره غیرهاب بررسی می‌شود اگر هاب ٦ به آن نزدیک‌تر است، هاب ٦ جایگزین هاب فعلی آن گره غیرهاب می‌شود. در این مثال، گره ۱ نیز به هاب ٦ تخصیص‌یافته است.

(٤) حذف هاب (*remove hub*): یک هاب را به‌طور تصادفی انتخاب و آن را گره غیرهاب می‌کند. گره‌های غیرهاب متصل به آن هاب بین هاب‌های باقیمانده توزیع می‌شوند. هر گره غیرهاب به نزدیک‌ترین هاب در جواب جدید تخصیص می‌یابد. به‌عنوان‌مثال، در **شکل ۵ (ب)** می‌خواهیم هاب ۸ را حذف کنیم، ابتدا باید گره‌های غیرهاب ٦ و ۷ که به هاب ۸ تخصیص‌یافته‌اند و خود هاب ۸ مجدداً تخصیص یابند. اکنون گره ۲ و ٤ هاب هستند؛ بنابراین باید گره‌های ٦، ۷ و ۸ بر اساس فاصله به هاب ۲ یا ٤ تخصیص یابند.

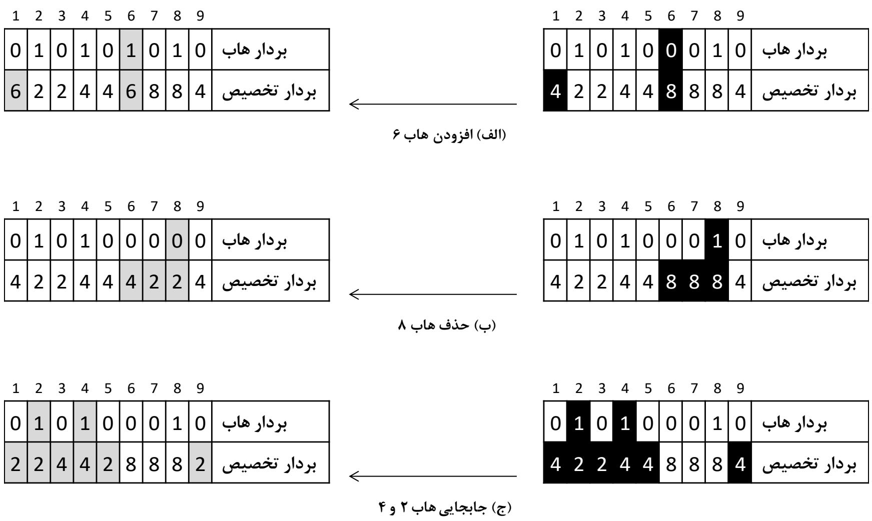
(۵) جابجایی هاب‌ها (*exchange hubs*): دو هاب را به‌طور تصادفی انتخاب می‌کند و گره‌های غیرهاب آن‌ها را جابجا می‌کند. در **شکل۵ (ج)** گره هاب ۲ و گره هاب ٤ انتخاب‌شده‌اند و گره‌های غیرهاب تخصیص‌یافته به آن‌ها جابجا شده‌اند.

برای هر عنصر جمعیت که تحت جهش قرار می‌گیرد، عملگرهای فوق اعمال‌شده‌اند و سپس نتیجه بهتر به‌عنوان جایگزین انتخاب‌شده است. این عملگرهای جهش جواب را از حالت قابل‌قبول خارج نمی‌کنند، به همین دلیل به فرآیند تعمیر نیازی ندارند. عملگرهای جهش شماره ۱ و ۲ درروش پیشنهادی تنها بر روی بردار تخصیص تغییراتی ایجاد می‌کنند. این عملگرهای جهش قبلاً درروش پیشنهادی (تاپکواوغلو و همکاران، ۲۰۰۵) مطرح‌شده‌اند. عملگر جهش شماره ۳، ٤ و ۵ درروش پیشنهادی بر روی هر دو بردار هاب و بردار تخصیص درراه حل تغییراتی ایجاد می‌کنند. این عملگرهای جهش نسبت به روش پیشنهادی (تاپکواوغلو و همکاران، ۲۰۰۵) جدید هستند و باعث افزایش قابلیت اکتشاف روش پیشنهادی می‌شوند.



شکل ۴- عملگرهای جهش گره‌های غیرهاب.

Figure 4- Spoke mutation operators.

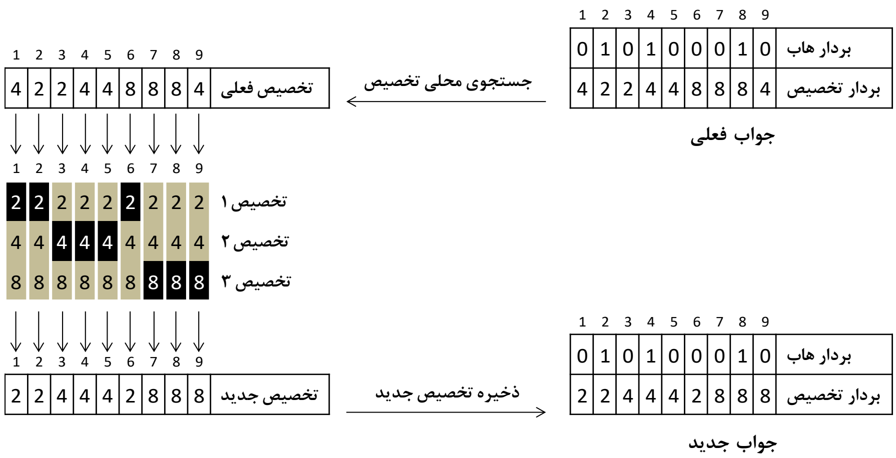


شکل ۵- عملگرهای جهش هاب و توزیع مجدد گره‌های غیرهاب.

Figure 5- Hub mutation operators and arrangement of spokes.

۴-۳- جستجوی محلی

در الگوریتم‌های تقلیدی[1] یک روش جستجوی محلی[2] با یک روش فرا ابتکاری ترکیب می‌شود. این روش‌ها از جستجوی محلی استفاده می‌کنند تا الگوریتم فرا ابتکاری را به سمت مناطق بهتر فضای جستجو هدایت کنند (ماریک و همکاران، ۲۰۱۳). در این مقاله، ما از جستجوی محلی برای جستجوی جوابی بهتر در اطراف بهترین جواب نسل جدید استفاده می‌کنیم. درصورتی‌که جواب بهتری پیدا شود، آن جواب جایگزین بهترین جواب نسل جدید می‌شود. این کار می‌تواند باعث بهبود سرعت همگرایی الگوریتم شود.



شکل ۶- جستجوی محلی تخصیص و قدم‌زنی حریصانه برای نزدیک شدن به جواب بهینه.

Figure 6- Allocation local search and greedy walk to approach the optimal solution.

ما جستجوی محلی را درروش پیشنهادی تنها بر روی بردار تخصیص بهترین جواب اعمال می‌کنیم و مجموعه هاب‌ها را در آن ثابت در نظر می‌گیریم. ابتدا گره‌های هاب و گره‌های غیرهاب بر اساس بردار هاب و بردار تخصیص کروموزوم جاری تعیین می‌شوند. سپس برای هر گره غیرهاب هزینه تخصیص به هر گره هاب موجود به ترتیب بررسی می‌شود. اگر تخصیص گره غیرهاب فعلی به گره هاب جدید باعث کاهش هزینه جواب شود، آن تخصیص پذیرفته‌شده و بر روی جواب فعلی اعمال می‌شود. همین روال در جستجوی محلی با بررسی تخصیص گره غیرهاب بعدی ادامه می‌یابد و هر بار که تخصیص جدید باعث کاهش هزینه جواب گردد، تخصیص جدید جایگزین تخصیص فعلی در جواب فعلی می‌شود. به‌این‌ترتیب ما سعی می‌کنیم در یک توالی از گره یک تا گره n تخصیص فعلی گره غیرهاب i را طوری اصلاح کنیم که هزینه کل جواب کاهش یابد. به‌عبارت‌دیگر، در یک اجرا از جستجوی محلی ممکن است تخصیص چند گره غیرهاب اصلاح‌شده و در چند قدم از جواب فعلی به جواب بهینه نزدیک‌تر شویم. فرآیند جستجوی محلی و قدم‌زنی حریصانه به سمت بهترین جواب در **شکل ٦** نمایش داده‌شده است.

باوجوداینکه جستجوی محلی یک فرآیند زمان‌بر است، استفاده از آن می‌تواند باعث دستیابی سریع‌تر الگوریتم به جواب‌های نزدیک بهینه‌شده و درمجموع تعداد تکرارهای الگوریتم و زمانی مصرفی را کاهش دهد. در این مقاله ما جستجوی محلی را تنها بر روی بهترین جواب تکرار فعلی اعمال کرده و به‌صورت تصادفی در بعضی تکرارها از آن استفاده می‌کنیم. برای این منظور یک عدد تصادفی در پایان هر تکرار تولید کرده و آن را با شانس جستجوی محلی مقایسه می‌کنیم. جستجوی محلی را در پایان تکرارهایی انجام می‌دهیم که عدد تصادفی تولیدشده کوچک‌تر از شانس تعیین‌شده برای جستجوی محلی باشد.

۴- یافته‌های پژوهش

در این بخش ما نتایج الگوریتم فرا ابتکاری پیشنهادی خود را برای حل مسئله مکان‌یابی هاب با تخصیص تکی ارائه می‌کنیم. روش پیشنهادی در *MATLAB R2015b* پیاده‌سازی شد. اجرای الگوریتم روی یک کامپیوتر با پردازنده *Intel Core2 Duo P8700 2.53 GHZ*

---

[1] Memetic algorithm

[2] Local search

و حافظه اصلی *4 GB RAM* انجام شد. برای ارزیابی الگوریتم از نمونه مسائل مجموعه داده پست استرالیا (*AP*) استفاده شد. روش پیشنهادی بر روی هر نمونه مسئله ۲۰ بار اجرا شد و برای هر نمونه مسئله میانگین زمان پردازش *CPUt* (برحسب ثانیه)، میانگین هزینه جواب به‌دست‌آمده *.Average.Sol* بهترین هزینه به‌دست‌آمده *.Best.Sol* میانگین فاصله هزینه جواب‌ها تا هزینه جواب بهینه *Average.Gap*؛ و قابلیت اعتماد *.Rel* (نرخ دستیابی به جواب بهینه) در الگوریتم ژنتیک ثبت شد. معیار فاصله تا جواب بهینه برای هر جواب به‌صورت *.Gap=(Solution.Cost.-Optimal.Cost.)/Optimal.Cost* محاسبه شد (گوآن و همکاران، ۲۰۱۸). در مورد نمونه مسائلی که جواب بهینه در دسترس نیست، مشابه سایر محققین به‌جای هزینه جواب بهینه از هزینه بهترین جوابی که تاکنون در مطالعات قبلی گزارش‌شده استفاده شد (گوآن و همکاران، ۲۰۱۸).

۴-۱- معرفی مجموعه داده

مجموعه مسائل *AP* مربوط به یک مسئله مکان‌یابی هاب در دنیای واقعی است که داده‌های آن به یک شرکت پستی در استرالیا تعلق دارد. این مجموعه داده اولین بار در مقاله (ارنست و کریشنامورتی، ۱۹۹۶) مطرح شد و از مجموعه *OR-Library* قابل دریافت است (باسلی[1]، ۱۹۹۰). این مجموعه داده شامل نمونه مسائلی با ۱۰، ۲۰، ۲۵، ٤۰، ۵۰، ۱۰۰ و ۲۰۰ گره است. داده‌های مجموعه پست استرالیا (*AP*) برای هر نمونه مسئله شامل هزینه ثابت هر گره، مکان گره‌ها در صفحه دوبعدی و میزان ترافیکی است که باید بین گره‌ها انتقال پیدا کند. مسائل مجموعه داده *AP* به دودسته سهل و سخت تقسیم می‌شوند. در نمونه مسائل سخت هزینه ثابت برای گره‌هایی با ترافیک بالا نسبت به مسائل سهل بزرگ‌تر در نظر گرفته‌شده است. به همین دلیل، در نمونه مسائل سخت انتخاب گره‌های هاب مشکل‌تر است و به جستجوی بیشتری نیاز دارد. در تمام نمونه مسائل این مجموعه داده مقدار ضرایب (α,γ,δ) به ترتیب برابر با (3,0.75,2) است.

۴-۲- تنظیم پارامترها

روش پیشنهادی ما دارای پنج پارامتر نرخ جهش (p_{mute})، نرخ تقاطع (p_{cross})، نرخ جستجوی محلی (p_{LS})، اندازه جمعیت (M) و حداکثر تعداد تکرارهای الگوریتم (I_{MAX}) است. در این بخش به‌منظور تعیین مقدار مناسب این پارامترها، مقادیر پیشنهادی برای هر پارامتر به‌صورت تک‌تک بررسی شد و بهترین مقدار هر پارامتر با توجه به معیار قابلیت اعتماد (*.Rel*) تعیین شد. مسئله سهل با اندازه ۲۰۰ گره از مجموعه داده پست استرالیا به‌عنوان نماینده سایر مسائل این مجموعه داده در نظر گرفته شد و تنظیم پارامترها به کمک آن انجام شد. برای افزایش دشواری این مسئله، هزینه ثابت گره‌های هاب در ضریب ۰/۸ ضرب شدند. مقدار پیش‌فرض نرخ تقاطع ۰/۹۰، نرخ جهش ۰/۲۰، اندازه جمعیت ۳۰۰ عضو، حداکثر تعداد تکرارها ۵۰۰ تکرار و نرخ جستجوی محلی صفر در نظر گرفته شد. ابتدا مقدار مناسب نرخ تقاطع مورد ارزیابی قرار گرفت که قابلیت اعتماد به‌دست‌آمده به ازای هر مقدار نرخ تقاطع در اولین نمودار میله‌ای **شکل ۷** آمده است. به ازای هر مقدار از نرخ تقاطع، روش پیشنهادی ۲۰ بار اجرا و میزان قابلیت اعتماد رسم شد. با توجه به این‌که عملگر تقاطع اصلی‌ترین عملگر الگوریتم ژنتیک است و امکان ساخت جواب‌های جدید را بر اساس جواب‌های قبلی فراهم می‌آورد، نرخ تقاطع روی مقدار ۰/۹۰ تنظیم شد. به همین ترتیب قابلیت اعتماد روش پیشنهادی در مقادیر مختلف نرخ جهش، اندازه جمعیت و نرخ جستجوی محلی ارزیابی و نمودار میله‌ای متناظر با هر پارامتر در **شکل ۷** رسم شد. بر اساس نتایج به‌دست‌آمده، در آزمایش‌های بخش‌های بعدی از مقدار $p_{cross} = 0.9$ برای نرخ تقاطع، $p_{mute} = 0.15$ برای نرخ جهش، $M = 200$ برای اندازه جمعیت، $I_{MAX} = 300$ برای حداکثر تعداد تکرارها و $P_{LS} = 0.05$ برای نرخ جستجوی محلی استفاده شد.

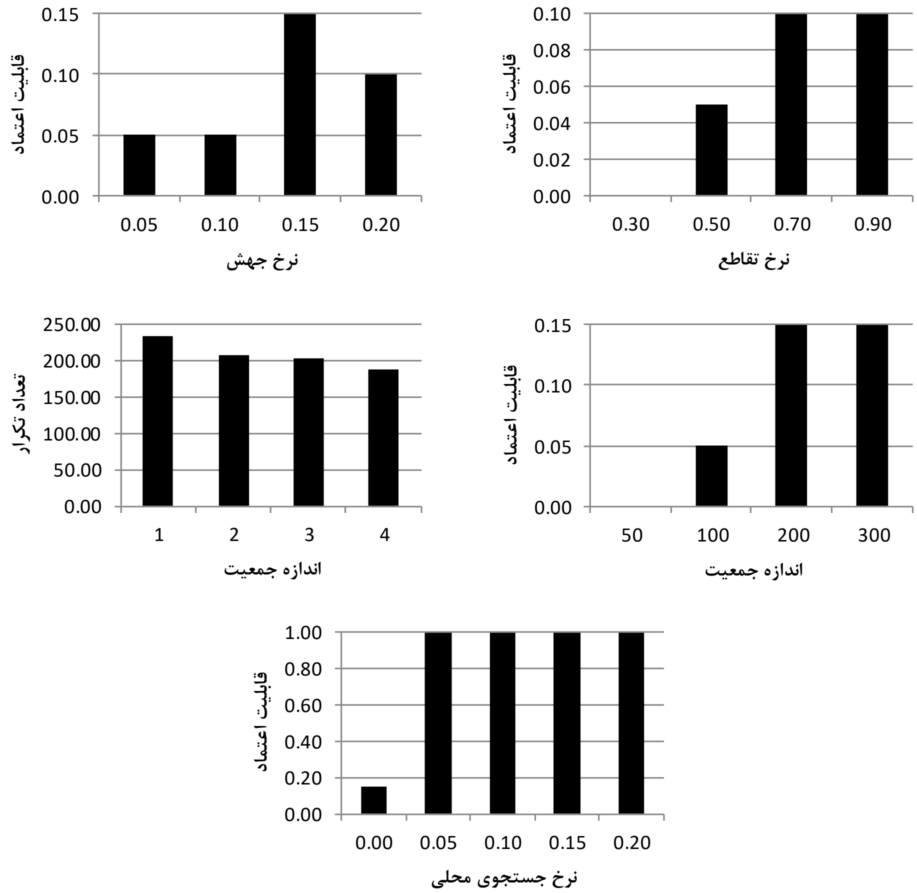
۴-۳- بررسی اثر عملگرهای مختلف

به‌منظور تبیین اثربخشی عملگرهای جهش هاب و عملگر جستجوی محلی درروش پیشنهادی، عملکرد روش پیشنهادی با و بدون این عملگرها ارزیابی شد. برای این منظور از دو نمونه مسئله سخت و سهل با ۱۰۰ گره در مجموعه داده پست استرالیا استفاده شد. نتیجه این ارزیابی در **جدول ۲** خلاصه‌شده است. هنگامی‌که عملگرهای جهش هاب و جستجوی محلی غیرفعال شدند، عملکرد الگوریتم

[1] Beasley

ژنتیک بسیار ضعیف بود. در این حالت الگوریتم ژنتیک برای مسئله سهل تنها در ۱۵% اجراها به جواب بهینه دست‌یافت و برای مسئله سخت در هیچ‌یک از اجراها به جواب بهینه نرسید.

شکل ۷- تنظیم مقدار پارامترهای مختلف الگوریتم پیشنهادی.
Figure 7- Tuning parameter values of the proposed algorithm.



با فعال شدن عملگرهای جهش هاب، کارآمدی الگوریتم ژنتیک بهبود یافت. در این حالت، الگوریتم ژنتیک برای برای مسئله سهل در ۴۰% اجراها و برای مسئله سخت در ۱۰۰% اجراها به جواب بهینه دست‌یافت. فعال شدن عملگر جستجوی محلی در کنار عملگرهای جهش هاب به بهبود بیشتر کارآمدی الگوریتم ژنتیک منجر شد. در این حالت، روش پیشنهادی برای مسئله سهل در ۹۵% اجراها و برای مسئله سخت در ۱۰۰% اجراها به جواب بهینه دست‌یافت. ازنظر میانگین هزینه راه‌حل در ۲۰ اجرا نیز با فعال شدن تدریجی عملگرهای پیشنهادی این مقاله، میانگین هزینه جواب‌ها کاهش یافت. البته استفاده از عملگرهای پیشنهادی باعث افزایش زمان مصرفی الگوریتم ژنتیک شده است، اما میزان این افزایش قابل‌تحمل است.

جدول ۲- بررسی اثر استفاده از عملگرهای جهش هاب و عملگر جستجوی محلی درروش پیشنهادی.
Table 2- Investigating the effects of hub mutations and local search in the proposed method.

نمونه مسئله	معیار ارزیابی	بدون جهش هاب و بدون جستجوی محلی	بدون جستجوی محلی	روش پیشنهادی
100 L	Rel.	0.15	0.40	0.95
	Best.Sol.	238016.28	238016.28	238016.28
	Average.Sol.	239777.1795	238083.3205	238017.936
	CPUt	22.32	28.56	33.22
100 T	Rel.	0.00	1.00	1.00
	Best.Sol.	417458	305097.95	305097.95
	Average.Sol.	425013.1285	305097.95	305097.95
	CPUt	10.81	14.19	14.42

۴-۴- ارزیابی عملکرد روش پیشنهادی

به‌منظور ارزیابی جامع عملکرد روش پیشنهادی، در این بخش روش پیشنهادی بر روی کلیه مسائل مجموعه داده پست استرالیا (*AP*) به کار گرفته شد و نتایج آن با دیگر روش‌ها مقایسه شد. از بین نمونه مسائل مجموعه داده پست استرالیا، تنها برای مسائل با اندازه ۱۰۰ گره و کمتر جواب بهینه با حل دقیق توسط نرم‌افزار *GAMS* و حل‌کننده *CPLEX* قابل‌محاسبه است. برای مسائل با اندازه ۲۰۰ گره به دلیل نیاز به حافظه بسیار زیاد برای ساخت درخت شاخه و هرس در *CPLEX*، حل دقیق مسئله با حافظه *RAM* محدود ممکن نیست. از طرف دیگر برای این مسائل، ساخت درخت و هرس کردن آن نیز زمان زیادی خواهد برد. برای حل مسئله در *CPLEX* از مدل خطی ارائه‌شده توسط سیلوا در (سیلوا و کانها، ۲۰۰۹) استفاده شد، زیرا مدل غیرخطی **رابطه (۱)** برای حل دقیق ناکارآمد است. **جدول ۳** نتیجه حل نمونه مسائل با تعداد حداکثر ۱۰۰ گره را در *GAMS* با روش پیشنهادی مقایسه می‌کند.

روش پیشنهادی توانسته است برای اکثر مسائل کوچک، در تمامی اجراها به جواب بهینه مسئله دست یابد و نرخ قابلیت اعتماد آن برای ۱۱ نمونه مسئله از بین ۱۲ نمونه مسئله به ۱۰۰% رسیده است. زمان لازم برای حل دقیق مسئله در *GAMS* به دو بخش تفکیک‌شده است: ستون *CPUt CPLEX* زمان مصرف‌شده توسط حل‌کننده *CPLEX* را نشان می‌دهد و ستون *CPUt GAMS* کل زمان لازم برای ساخت مدل در گمز، فراخوانی حل‌کننده و سپس نمایش نتایج را نشان می‌دهد. مقایسه زمان لازم برای حل دقیق مسئله بازمان حل مسئله توسط روش پیشنهادی نشان می‌دهد که با افزایش اندازه مسئله الگوریتم ژنتیک پیشنهادی زمان بسیار کمتری را نسبت به حل دقیق نیاز دارد. به‌عنوان‌مثال، برای نمونه مسئله سخت با اندازه ۱۰۰ گره روش پیشنهادی تنها به ١٤/٤٢ ثانیه زمان نیاز دارد. در مقابل، حل دقیق در گمز به ١٤٤/٦٩ ثانیه زمان نیاز داشته که ۹۹/۸۸ ثانیه آن توسط حل‌کننده *CPLEX* مصرف‌شده است. علاوه بر زمان مصرفی بالا، حل دقیق به حافظه *RAM* قابل‌توجهی هم نیاز دارد. در آزمایش انجام‌شده حل دقیق نمونه مسئله با ۱۰۰ گره در گمز به ٤٧١ مگابایت و حل نمونه مسئله با ۲۰۰ گره به بیش از ٤ گیگابایت حافظه *RAM* نیاز دارد. این در حالی است که حافظه مصرفی روش پیشنهادی برای همین نمونه مسائل کمتر از ۵ مگابایت است. این موضوع باعث می‌شود تا حل دقیق مسائل بزرگ‌تر از ۱۰۰ گره در سیستم‌های کامپیوتری فعلی ممکن نباشد. حتی در صورت وجود حافظه کافی، زمان مصرفی حل دقیق که بخش اعظم آن به ساخت درخت و سپس هرس درخت مربوط است، برای نمونه مسائل بزرگ غیرقابل‌تحمل خواهد بود.

جدول ۳- مقایسه عملکرد روش پیشنهادی در حل نمونه مسائل مجموعه داده پست استرالیا با حل بهینه در GAMS.

Table 3- Performance comparison between proposed method and GAMS on AP dataset.

مسئله	GAMS & CPLEX					روش پیشنهادی					
اندازه	Optimal. Hubs.	Optimal. Sol.	CPUt CPLEX	CPUt GAMS	Memory GAMS	Best. Hubs.	Best. Sol.	Average. Sol.	Average. Gap.	CPUt	Rel.
10 L	3,4,7	224250.06	0.44	4.02	4 MB	3,4,7	224250.06	224250.06	0.0000	1.22	1.00
20 L	7,14	234690.94	0.22	4.93	8 MB	7,14	234690.96	234690.96	0.0000	2.01	1.00
25 L	8,18	236650.60	0.50	4.71	11 MB	8,18	236650.63	236650.63	0.0000	2.59	1.00
40 L	14,28	240986.23	10.70	17.04	34 MB	14,28	240986.23	240986.23	0.0000	5.01	1.00

1.00	8.62	0.0000	237421.99	237421.99	15,36	62 MB	21.40	12.80	237421.98	15,36	50 L
0.95	33.22	0.0000	238017.94	238016.28	29,73	471 MB	322.89	279.34	238016.28	29,73	100 L
1.00	1.37	0.0000	263399.95	263399.95	4,5,10	4 MB	6.80	0.14	263399.92	4,5,10	10 T
1.00	2.50	0.0000	271128.18	271128.18	7,19	8 MB	4.49	0.27	271128.14	7,19	20 T
1.00	2.41	0.0000	295667.84	295667.84	13	11 MB	4.64	0.28	295667.83	13	25 T
1.00	3.98	0.0000	293164.84	293164.84	19	34 MB	7.51	1.16	293164.83	19	40 T
1.00	5.57	0.0000	300420.99	300420.99	24	62 MB	17.93	9.20	300420.98	24	50 T
1.00	14.42	0.0000	305097.95	305097.95	52	471 MB	144.69	99.88	305097.95	52	100 T

روش پیشنهادی این مطالعه توسعه‌ای بر الگوریتم ژنتیک پیشنهادی تاپکواوغلو در (تاپکواوغلو و همکاران، ۲۰۰۵) برای حل مسئله مکان یابی بدون ظرفیت هاب با تخصیص تکی است. تاپکواوغلو در (تاپکواوغلو و همکاران، ۲۰۰۵) عملکرد الگوریتم ژنتیک پیشنهادی خود را بر روی مسائل مجموعه داده پست استرالیا ارزیابی کرده است. داده‌های خام مجموعه داده پست استرالیا برای موقعیت مکانی گره‌ها شامل موقعیت ۲۰۰ گره است که برای مسائل کوچک‌تر از ۲۰۰ گره باید موقعیت گره‌ها به کمک یک قطعه کد به زبان C از روی موقعیت ۲۰۰ گره اصلی محاسبه و درون‌یابی شود. تاپکواوغلو در (تاپکواوغلو و همکاران، ۲۰۰۵) برای مسئله با اندازه ۲۰۰ گره نیز سهواً به‌جای استفاده از موقعیت‌های خام ارائه‌شده، موقعیت گره‌ها را درون‌یابی کرده است. به همین دلیل هزینه بهترین جواب مسئله با اندازه ۲۰۰ گره در مقاله تاپکواوغلو با منابع جدیدتر مثل (سیلوا و کانها، ۲۰۰۹) و (گوآن و همکاران، ۲۰۱۸) متفاوت است. برای تأکید بیشتر بر این موضوع نمونه مسئله درون‌یابی شده با اندازه ۲۰۰ گره را در نتایج با *۲۰۰ نشان می‌دهیم. به‌منظور ارزیابی اثربخشی بهبودهای پیشنهادی ما بر الگوریتم پیشنهادی تاپکواوغلو، عملکرد روش پیشنهادی روی نمونه مسائل استفاده‌شده توسط تاپکواوغلو ارزیابی شد. **جدول ٤** نتایج روش پیشنهادی را با نتایج گزارش‌شده توسط تاپکواوغلو در (تاپکواوغلو و همکاران، ۲۰۰۵) مقایسه می‌کند. بررسی نتایج در **جدول ٤** نشان می‌دهد که قابلیت اعتماد روش پیشنهادی ما با روش تاپکواوغلو تفاوت چشمگیری دارد. روش پیشنهادی ما با اکتشاف بهتر فضای جستجو توانسته است برای اکثر نمونه مسائل مجموعه داده پست استرالیا در تمام ۲۰ اجرا به جواب بهینه دست یابد و به نرخ قابلیت اعتماد ١٠٠% در اکثر مسائل رسیده است. این در حالی است که روش تاپکواوغلو برای نمونه مسئله سخت با ٤٠ گره تنها در ١٠% اجراها و برای نمونه مسئله سخت با ۵۰ گره تنها در ٤% اجراها به جواب بهینه دست‌یافته است. در مورد مسائل بزرگ‌تر عملکرد روش تاپکواوغلو بازهم ضعیف‌تر بوده و برای مسئله سهل با ۱۰۰ گره تنها در ۲% اجراها و برای مسئله سهل با ۲۰۰ گره تنها در ۱% اجراها به جواب بهینه دست یافته است. در مقابل، روش پیشنهادی ما برای مسئله سهل با ۱۰۰ گره در ۹۵% اجراها و برای مسئله سهل با ۲۰۰ گره در ۸۵% اجراها به جواب بهینه دست‌یافته است. لذا جواب ارائه‌شده توسط الگوریتم پیشنهادی نسبت به جواب ارائه‌شده توسط الگوریتم تاپکواوغلو در یک اجرا قابل‌اعتمادتر است و نیازی به چند بار اجرای الگوریتم پیشنهادی بر روی هر نمونه مسئله احساس نمی‌شود. در **جدول ٤** زمان مصرفی روش تاپکواوغلو با زبان برنامه‌نویسی متفاوت و بر روی سیستم کامپیوتری متفاوتی نسبت به مطالعه ما ارزیابی‌شده است. به همین دلیل مقایسه مستقیم زمان‌های مصرفی منصفانه نیست و باید از آن خودداری شود. این نتایج نشان می‌دهند که بهبودهای انجام‌شده با مطرح کردن عملگرهای جستجوی جدید درروش پیشنهادی مؤثر بوده و کارآمدی الگوریتم ژنتیک را افزایش داده است. شایان‌ذکر است که تفاوت‌های جزئی در هزینه بهترین راه‌حل درروش تاپکواوغلو برای هر مسئله نسبت به راه‌حل بهینه به‌دست‌آمده با گمز و بهترین راه‌حل محاسبه‌شده توسط روش پیشنهادی ما می‌تواند ناشی از جزئیات محاسبه فاصله گره‌ها و دقت اعداد اعشاری در آزمایش‌های تاپکواوغلو باشد.

جدول ۴- مقایسه عملکرد روش پیشنهادی در حل نمونه مسائل مجموعه داده پست استرالیا با روش تاپکواوغلو.

Table 4- Performance comparison between proposed method and Topcuoglu's on AP dataset.

روش تاپکواوغلو				روش پیشنهادی						مسئله
Rel.	CPUt	Best. Sol.	Best. Hubs.	Rel.	CPUt	Average. Gap.	Average. Sol.	Best. Sol.	Best. Hubs.	اندازه
0.83	0.22	224249.82	3,4,7	1.00	1.22	0.0000	224250.06	224250.06	3,4,7	10 L
0.96	0.92	234690.11	7,14	1.00	2.01	0.0000	234690.96	234690.96	7,14	20 L
0.90	1.47	236649.69	8,18	1.00	2.59	0.0000	236650.63	236650.63	8,18	25 L

0.55	4.79	240985.51	14,28	1.00	5.01	0.0000	240986.23	240986.23	14,28	40 L	
0.59	8.32	237420.69	15,36	1.00	8.62	0.0000	237421.99	237421.99	15,36	50 L	
0.02	54.23	238017.53	29,73	0.95	33.22	0.0000	238017.94	238016.28	29,73	100 L	
0.01	439.08	228973.67	53,184	0.85	83.76	0.0010	229171.87	228947.08	53,184	200* L	
0.66	0.18	263402.13	4,5,10	1.00	1.37	0.0000	263399.95	263399.95	4,5,10	10 T	
0.28	0.71	271128.41	7,19	1.00	2.50	0.0000	271128.18	271128.18	7,19	20 T	
1.00	1.14	295670.39	13	1.00	2.41	0.0000	295667.84	295667.84	13	25 T	یکی
0.10	3.49	293163.38	19	1.00	3.98	0.0000	293164.84	293164.84	19	40 T	از
0.04	5.82	300420.87	24	1.00	5.57	0.0000	300420.99	300420.99	24	50 T	
0.01	39.31	305101.07	29,73	1.00	14.42	0.0000	305097.95	305097.95	52	100 T	
0.02	415.82	233570.44	52	0.90	110.01	0.0001	233551.77	233540.24	53,184	200* T	

جدیدترین مطالعات درزمینهٔ ارائه یک الگوریتم جدید برای حل مسئله مکان‌یابی بدون ظرفیت هاب با تخصیص تکی، روش پیشنهادی گوآن و همکاران (۲۰۱۸) است. این روش یک الگوریتم جستجوی محلی تکراری چند شروعه[1] (*MSLSA*) است که از مکانیزم شروع مجدد برای رهایی از بهینه محلی استفاده می‌کند. **جدول ۵** نتایج روش پیشنهادی را با نتایج گزارش‌شده توسط گوآن در مقاله (گوآن و همکاران، ۲۰۱۸) مقایسه می‌کند. ازنظر معیار هزینه‌ی بهترین جواب (*Best.Sol.*) روش پیشنهادی ما به هزینه‌ی مشابه با روش گوآن بر روی مسائل تا اندازه ۲۰۰ گره دست‌یافته است. ازنظر معیار میانگین فاصله تا جواب بهینه (*Average.Gap.*) نیز هر دو روش بر روی مسائلی تا اندازه ۲۰۰ گره به فاصله نزدیک به صفر دست‌یافته‌اند. زمان مصرفی روش گوآن نیز با زبان برنامه‌نویسی متفاوت و بر روی سیستم کامپیوتری متفاوتی نسبت به مطالعه ما ارزیابی‌شده است. به همین دلیل مقایسه مستقیم زمان‌های مصرفی منصفانه نیست و از آن خودداری شد. روش پیشنهادی گوآن یک جستجوی تک نقطه‌ای است و همواره با یک جواب فعلی سروکار دارد. درحالی‌که الگوریتم ژنتیک یک جستجوی چندنقطه‌ای مبتنی بر جمعیت است. از طرف دیگر، درروش پیشنهادی ما جستجوی محلی روی کلیه همسایه‌های بهترین جواب انجام می‌شود که باعث شده با افزایش اندازه مسئله عملگر جستجوی محلی پرهزینه و زمان‌بر باشد. در مقابل، گوآن جستجوی محلی را بر روی بعضی همسایه‌های جواب فعلی و به‌تصادف انجام داده است. این تفاوت‌ها نیز در سرعت‌بالای روش پیشنهادی گوآن مؤثر است.

جدول ۵- مقایسه عملکرد روش پیشنهادی در حل نمونه مسائل مجموعه داده پست استرالیا با روش گوآن.

Table 5- Performance comparison between proposed method and Guan's on AP dataset.

روش MSLSA گوآن			روش پیشنهادی						مسئله
CPUt	**Average. Gap.**	**Best. Sol.**	**Rel.**	**CPUt**	**Average. Gap.**	**Average. Sol.**	**Best. Sol.**	**Best. Hubs.**	**اندازه**
0.00	0.00	224250.05	1.00	1.22	0.0000	224250.06	224250.06	3,4,7	10 L
0.00	0.00	234690.95	1.00	2.01	0.0000	234690.96	234690.96	7,14	20 L
0.01	0.00	236650.62	1.00	2.59	0.0000	236650.63	236650.63	8,18	25 L
0.03	0.00	240986.23	1.00	5.01	0.0000	240986.23	240986.23	14,28	40 L
0.07	0.00	237421.98	1.00	8.62	0.0000	237421.99	237421.99	15,36	50 L
0.43	0.00	238016.28	0.95	33.22	0.0000	238017.94	238016.28	29,73	100 L
3.61	0.00	233802.98	0.90	183.28	0.0000	233806.83	233802.98	43,148	200 L
0.00	0.00	263399.94	1.00	1.37	0.0000	263399.95	263399.95	4,5,10	10 T
0.01	0.00	271128.18	1.00	2.50	0.0000	271128.18	271128.18	7,19	20 T
0.01	0.00	295667.84	1.00	2.41	0.0000	295667.84	295667.84	13	25 T
0.03	0.00	293164.84	1.00	3.98	0.0000	293164.84	293164.84	19	40 T
0.06	0.00	300420.98	1.00	5.57	0.0000	300420.99	300420.99	24	50 T
0.48	0.00	305097.96	1.00	14.42	0.0000	305097.95	305097.95	52	100 T

[1] Multi-start Local Search Algorithm (MSLSA)

3.74	0.00	272188.11	0.45	162.10	0.0013	272548.86	272188.11	54,122	200 T

۵- بحث و نتیجه‌گیری

در این مقاله یک الگوریتم ژنتیک بهبودیافته برای حل مسئله مکان‌یابی هاب با تخصیص تکی در شبکه‌های حمل‌ونقل ارائه شد. الگوریتم ژنتیک کار خود را از یک جمعیت اولیه از جواب‌ها آغاز کرده و با کمک ترکیب جواب‌ها و جهش‌های تصادفی سعی می‌کند تا کیفیت جواب‌ها را بهبود دهد. نتیجه اجرای الگوریتم ژنتیک به جمعیت اولیه و عملگرهای ژنتیک مورداستفاده بستگی دارد. در صورت عدم استفاده از عملگرهای جهش مناسب، گوناگونی جمعیت به‌تدریج کاهش‌یافته و الگوریتم ژنتیک در بهینه محلی گرفتار خواهد شد. در این صورت نتیجه اجرای الگوریتم ژنتیک به کیفیت جمعیت اولیه وابستگی زیادی خواهد داشت و الگوریتم تنها در بعضی اجراها به جواب مطلوب دست خواهد یافت. به همین دلیل درروش پیشنهادی عملگرهای جهش هاب ازجمله افزودن هاب، حذف هاب و تعویض هاب با غیرهاب برای فرار از بهینه محلی و ایجاد گوناگونی بیشتر به روش پایه اضافه شدند. در کنار عملگرهای جهش، یک عملگر جستجوی محلی در اطراف بهترین جواب هر تکرار نیز مطرح شد تا الگوریتم سریع‌تر به ناحیه امیدوارکننده فضای جستجو نزدیک شود. بهبودهای انجام‌شده درروش پیشنهادی باعث شد تا گوناگونی جمعیت در تکرارهای متوالی در سطح خوبی حفظ شود و قابلیت اکتشاف الگوریتم بالا باشد. درنتیجه، وابستگی الگوریتم ژنتیک به کیفیت جمعیت اولیه کاهش یافت. روش پیشنهادی این توانمندی را دارد که حتی با شروع از یک جمعیت اولیه ضعیف ناحیه امیدوارکننده فضای جستجو را کشف کند و به جواب بهینه همگرا شود. در مورد مسائل کوچک و متوسط، الگوریتم پیشنهادی نسبت به حل دقیق مسئله در *GAMS* زمان مصرفی را به یک‌دهم زمان *GAMS* کاهش داد. روش پیشنهادی توانایی حل مسائل بزرگ‌تری را نسبت به *GAMS* دارد. حل دقیق مسئله‌ای با ۲۰۰ گره در *GAMS* به دلیل نیاز به حافظه زیاد ممکن نشد، درحالی‌که الگوریتم پیشنهادی به جواب مسئله دست‌یافت. در مقایسه با روش پایه تاپکواوغلو، روش ما به کمک عملگرهای اضافه‌شده توانست در تعداد بیشتری از اجراها به جواب بهینه هر نمونه مسئله همگرا شود. درنهایت، روش ما در مقایسه با روش گوآن که در سال ۲۰۱۸ مطرح‌شده است، ازنظر هزینه بهترین جواب و میانگین فاصله تا جواب بهینه عملکرد مشابهی را ارائه داد. حل نمونه مسائل بزرگ‌تر از ۲۰۰ گره در مسئله مکان‌یابی بدون ظرفیت هاب با تخصیص تکی با سرعتی بالا و قابلیت اعتماد بالا هنوز یک چالش مهم است. در این راستا، پیشنهاد الگوریتم جستجوی کارآمد، استفاده از عملگرهای جستجوی مناسب، اکتشاف کامل فضای جستجو و فرار از بهینه‌های محلی می‌تواند به یک الگوریتم مناسب بیانجامد. الگوریتمی برای حل مسئله مکان‌یابی هاب مناسب است که توانایی اکتشاف بالایی در فضای جستجو داشته باشد و موازنه خوبی بین اکتشاف و استخراج برقرار سازد. طراحی عملگرهای جستجوی همسایگی، عملگرهای جستجوی محلی و استفاده از عملگرهای آشفته سازی (*perturbation*) و لرزش (*shaking*) می‌تواند به فرار از بهینه محلی کمک کند. بر این اساس، طراحی عملگرهای مناسب برای حل مسئله مکان‌یابی هاب و استفاده از روش‌های فرا ابتکاری دیگر برای حل مسئله مکان‌یابی هاب ازجمله اهداف آتی محققین خواهد بود.

سپاسگزاری

نویسندگان مقاله مراتب قدردانی خود را از داوران محترم اعلام می‌دارند. بی‌شک نقطه نظرات ارزشمندشان در بهبود کیفیت مقاله نقش بسزایی داشته است.

تعارض با منافع

نویسندگان اعلام می‌دارند که هیچ تضادی در منافع در مورد انتشار این نسخه وجود ندارد.

منابع

Abdinnour-Helm, S. (1998). A hybrid heuristic for the uncapacitated hub location problem. *European journal of operational research*, *106*(2-3), 489-499. https://doi.org/10.1016/S0377-2217(97)00286-5

Abdinnour-Helm, S., & Venkataramanan, M. A. (1998). Solution approaches to hub location problems. *Annals of operations research*, *78*, 31-50. https://doi.org/10.1023/A:1018954217758

Abyazi-Sani, R., & Ghanbari, R. (2016). An efficient tabu search for solving the uncapacitated single allocation hub location problem. *Computers & industrial engineering*, *93*, 99-109. https://doi.org/10.1016/j.cie.2015.12.028

Akbaripour, H., Masehian, E., & Roostaei, A. (2017). Landscape analysis and scatter search metaheuristic for solving the uncapacitated single allocation hub location problem. *International journal of industrial and systems engineering*, *26*(4), 425-459. DOI: 10.1504/IJISE.2017.085207

Alumur, S., & Kara, B. Y. (2008). Network hub location problems: the state of the art. *European journal of operational research*, *190*(1), 1-21. https://doi.org/10.1016/j.ejor.2007.06.008

Bailey, A., Ornbuki-Berrnan, B., & Asobiela, S. (2013, April). Discrete pso for the uncapacitated single allocation hub location problem. *2013 IEEE symposium on computational intelligence in production and logistics systems (CIPLS)* (pp. 92-98). IEEE. DOI: 10.1109/CIPLS.2013.6595205

Bashiri, M., Karimi, H., Akhondzadeh, E., Dehghan, E., Karani, E., Ghasemi Nejad, A., Ramezani, M., Fotouhi, F., & Rajabali Kani, A. (2011). *Design of industrial systems II (second Ed.)*. Shahed University Press. (**In Persian**). https://www.adinehbook.com/gp/product/6006121093

Beasley, J. E. (1990). OR-Library: distributing test problems by electronic mail. *Journal of the operational research society*, *41*(11), 1069-1072. https://doi.org/10.1057/jors.1990.166

De Carvalho, R., Gomes, B., Martins, A., Saldanha, R., & de Camargo, R. (2017). A multi-start heuristic for the design of hub-and-spoke networks. *Proceeding Series of the Brazilian society of computational and applied mathematics*, *5*(1), 1-7. DOI: https://doi.org/10.5540/03.2017.005.01.0471

Ernst, A. T., & Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location science*, *4*(3), 139-154. https://doi.org/10.1016/S0966-8349(96)00011-3

Filipović, V., Kratica, J., Tošić, D., & Dugošija, D. (2009). GA inspired heuristic for uncapacitated single allocation hub location problem. In *Applications of soft computing* (pp. 149-158). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-89619-7_15

Ghaffarinasab, N., & Kara, B. Y. (2019). Benders decomposition algorithms for two variants of the single allocation hub location problem. *Networks and spatial economics*, *19*(1), 83-108. https://doi.org/10.1007/s11067-018-9424-z

Guan, J., Lin, G., & Feng, H. B. (2018a). A learning-based probabilistic tabu search for the uncapacitated single allocation hub location problem. *Computers & operations research*, *98*, 1-12. https://doi.org/10.1016/j.cor.2018.04.020

Guan, J., Lin, G., & Feng, H. B. (2018b). A multi-start iterated local search algorithm for the uncapacitated single allocation hub location problem. *Applied soft computing*, *73*, 230-241. https://doi.org/10.1016/j.asoc.2018.08.035

Hekmatfar, M., & Pishvaee, M. (2009). Hub location problem. In R. Zanjirani Farahani & M. Hekmatfar (Eds), *Facility Location* (pp. 243-270). Physica, Heidelberg. https://doi.org/10.1007/978-3-7908-2151-2_11

Marić, M., Stanimirović, Z., & Stanojević, P. (2013). An efficient memetic algorithm for the uncapacitated single allocation hub location problem. *Soft computing*, *17*(3), 445-466. https://doi.org/10.1007/s00500-012-0919-0

O'kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European journal of operational research*, *32*(3), 393-404. https://doi.org/10.1016/S0377-2217(87)80007-3

Rabbani, M., Farrokhi-Asl, H., & Heidari, R. (2017). Genetic algorithm-based optimization approach for an uncapacitated single allocation P-hub center problem with more realistic cost structure. *Journal of industrial and systems engineering*, *10*(1), 108-124.

Silva, M. R., & Cunha, C. B. (2009). New simple and efficient heuristics for the uncapacitated single allocation hub location problem. *Computers & operations research*, *36*(12), 3152-3165. https://doi.org/10.1016/j.cor.2008.12.019

Topcuoglu, H., Corut, F., Ermis, M., & Yilmaz, G. (2005). Solving the uncapacitated hub location problem using genetic algorithms. *Computers & operations research*, *32*(4), 967-984. https://doi.org/10.1016/j.cor.2003.09.008